Multivariate Statistics with R

Confirmatory Factor Analysis

Aja Murray

Aja.Murray@ed.ac.uk

This Week

- Techniques
  - *Confirmatory Factor Analysis (CFA)*
- Key Functions
  - *cfa( ) from lavaan package*
- Reading
  - *lavaan tutorial: http://lavaan.ugent.be/tutorial/tutorial.pdf (sections 1-4)*
  - *lavaan paper: https://www.jstatsoft.org/article/view/v048i02*

Learning Outcomes

- Know what it means to specify, estimate, and evaluate a CFA model
- Fit and interpret CFA models in R using the cfa( ) function
- Visualise CFA models using SEM diagrams

Overview of this lecture

- Introduction to CFA
- Model Specification
- Model Identification
- Model Estimation
- Model Evaluation
- Model Modification

Introduction to CFA

- Used to test a factor structure for a set of variables
- EFA is used when we have no fixed idea of our factor structure
- CFA is used to test a particular factor structure
- CFA tests how well our proposed factor structure fits the data
- Like EFA, CFA is a latent variable model
  - *observed variables serve as* ***indicators*** *of underlying latent factors*
- Unlike EFA, only specific loadings are included in the model

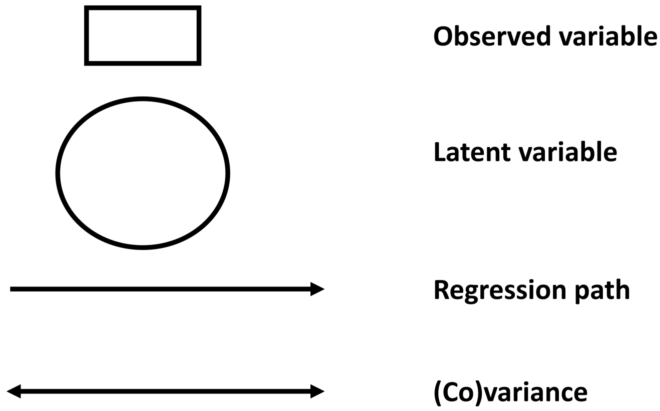
The variance-covariance matrix

- Our starting point for CFA is the variance-covariance matrix for our items
- CFA models represent these variances/covariances in terms of a set of latent factors

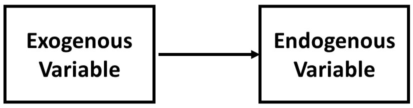
```
round(cov(agg.items),2)
```

```
##        item1 item2 item3 item4 item5 item6 item7 item8 item9 item10
## item1   0.91  0.50  0.46  0.43  0.53  0.06  0.13  0.09  0.10   0.07
## item2   0.50  0.99  0.59  0.52  0.64  0.02  0.12  0.08  0.09   0.04
## item3   0.46  0.59  0.97  0.46  0.60  0.05  0.11  0.08  0.14   0.03
## item4   0.43  0.52  0.46  0.96  0.55  0.06  0.14  0.11  0.09   0.06
## item5   0.53  0.64  0.60  0.55  0.96  0.01  0.11  0.05  0.11   0.01
## item6   0.06  0.02  0.05  0.06  0.01  0.99  0.53  0.52  0.40   0.40
## item7   0.13  0.12  0.11  0.14  0.11  0.53  0.93  0.73  0.55   0.56
## item8   0.09  0.08  0.08  0.11  0.05  0.52  0.73  0.93  0.55   0.59
## item9   0.10  0.09  0.14  0.09  0.11  0.40  0.55  0.55  0.98   0.44
## item10  0.07  0.04  0.03  0.06  0.01  0.40  0.56  0.59  0.44   0.95
```

SEM Diagrams

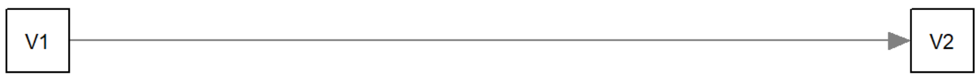


Exogenous versus endogeneous variables



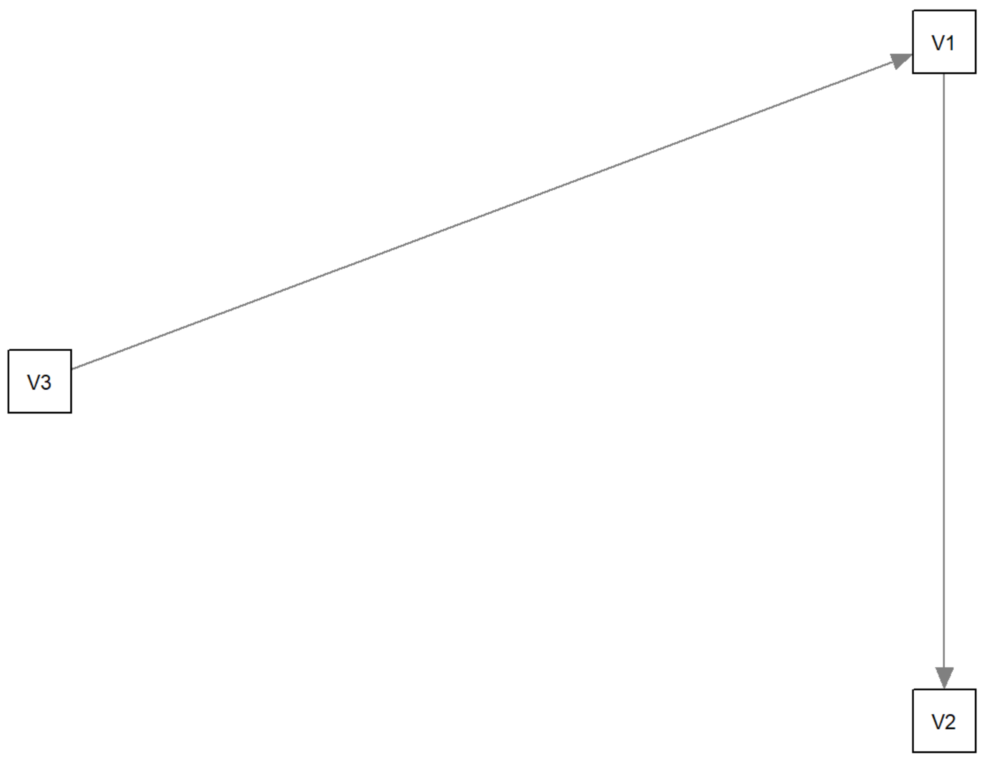
- **exogenous** variables receive input from no other variables
  - *they emanate but are not on the end of single-headed arrow paths*
  - *they are the ‘independent variables’ or ‘predictors’*
- **endogenous** variables receive input from other variables
  - *they are on the end of single-headed arrow paths*
  - *they are the ‘dependent variables’ or ‘outcomes’*
  - *they may also be predictors of other variables in the model*

SEM diagram for a simple regression model

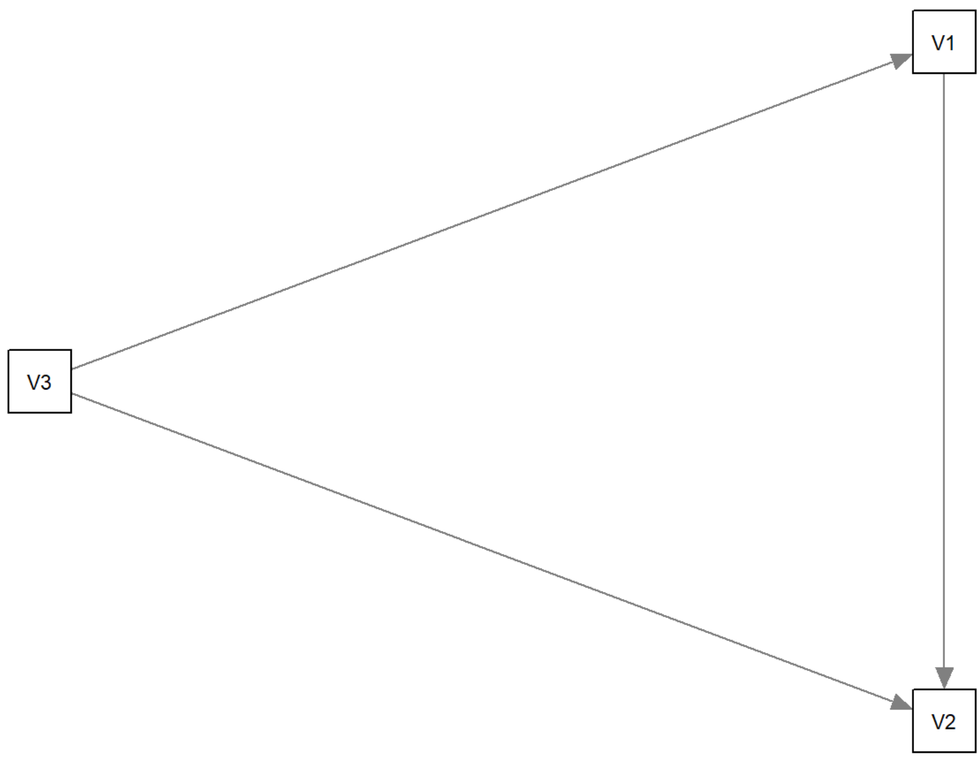


SEM diagram for a covariance

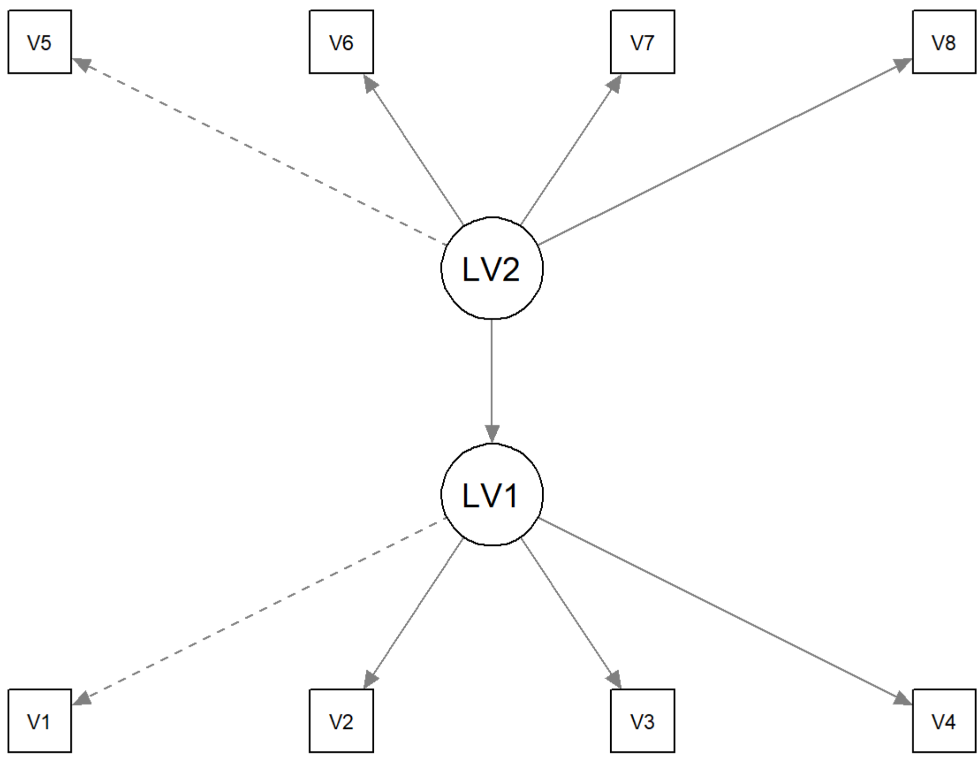
SEM diagram for a path analysis model



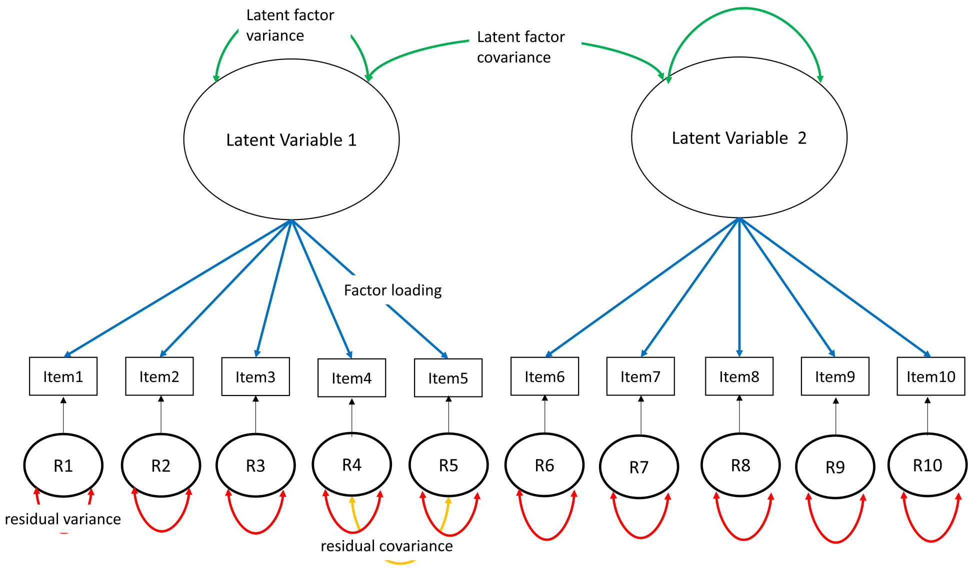
SEM diagram for another path analysis model



SEM diagram for a more complex model



The CFA model



The parameters of a CFA

- Latent factor variances and covariances
  - *The variability of and associations between the latent factors*
- Factor loadings
  - *Regression of the latent variables on the observed variables*
  - *Strength of relation between underlying latent variables and observed variables*
- Residual variances
  - *Variance in the observed variables not explained by the latent variables*
- Residual covariances
  - *The covariances between observed variables that exist over and above their covariance due to their shared relation with a latent factor*
- CFAs involve finding (or specifying) values for all of these parameters

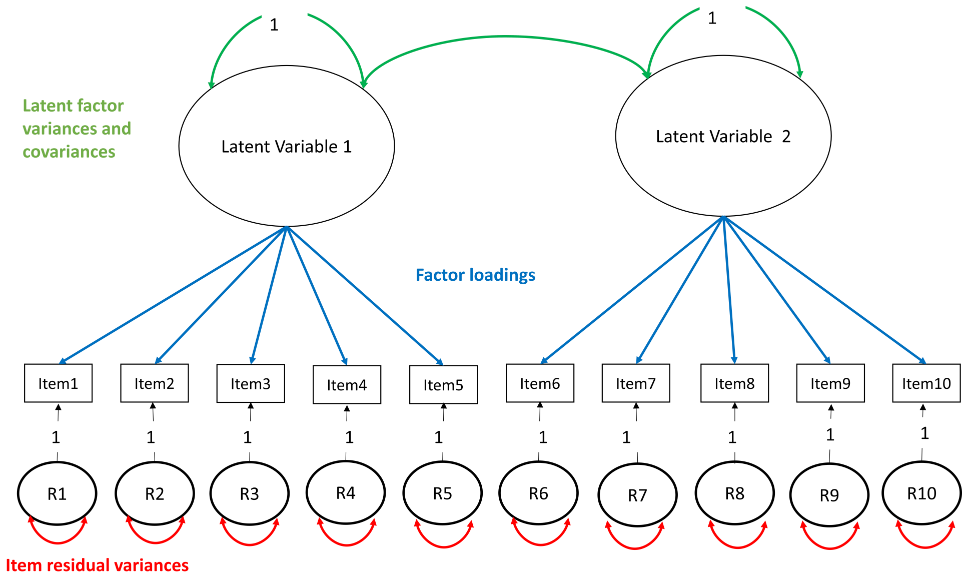
Model specification

- Defining the model we want to test
    - *i.e., which **parameters** do we want to estimate?*
        - How many factors?
        - Which items do we think go with which factors?
        - Are the factors correlated?
- Based on theory or past research (e.g., previous EFAs)

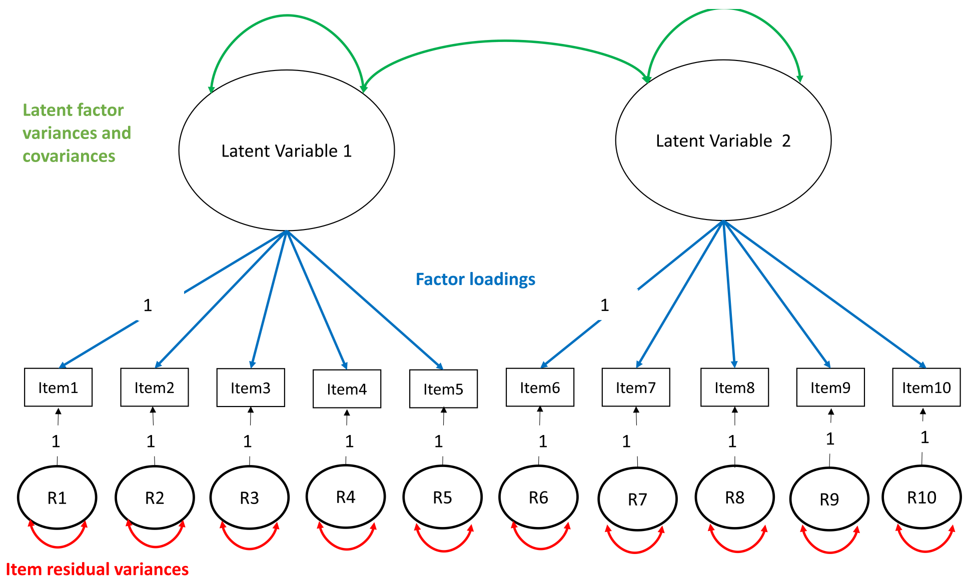
Latent variable scaling

- Latent variable scaling is a key aspect of model specification
- Latent variables have no inherent scale, so we have to define one
- Two commonly used scaling methods:
  - *Method 1: Fix the variance of each latent variable to 1*
  - *Method 2: Fix one factor loading for each latent variable to 1*
- Note that the necessity of scaling also applies to the residual factors
  - *Typically uses Method 2*

Scaling the latent variables by fixing factor variances



Scaling the latent variables by fixing factor loadings



Model identification

- More generally, we need to ensure that the model we specify is **identified**
- Identification concerns the number of ‘knowns’ versus ‘unknowns’
- There must be more knowns than unknowns in order to be able to test a CFA
- In CFA, the knowns are variances and covariances of the observed variables
- The unknowns are the parameters we want to estimate
- **Degrees of freedom** are the difference between knowns and unknowns

Levels of identification

- There are three levels of identification:
- **Under-identified** models: have < 0 degrees of freedom
- **Just Identified** models: have 0 degrees of freedom
- **Over-Identified** models: have > 0 degrees of freedom

Model identification illustration

- Chou & Bentler (1995) provide an illustration based on simultaneous linear equations:
  - *Eq.1:* $x + y = 5$
  - *Eq.2:* $2x + y = 8$
  - *Eq.3:* $x + 2y = 9$
- Eq.1 is on its own is *under-identified*
- Eq.1 & 2 are together *just identified*
- Eq.1, 2 & 3 are together *over identified*

The number of knowns

- To ensure model identification, we need to know the number of knowns
- We can calculate the knowns by:

$$\frac{(k+1)\,(k)}{2}$$

- where *k* is the number of observed variables.

The number of knowns

- This is the number of unique elements in the variance-covariance matrix for our observed variables
  - *e.g., if we had three observed variables:*

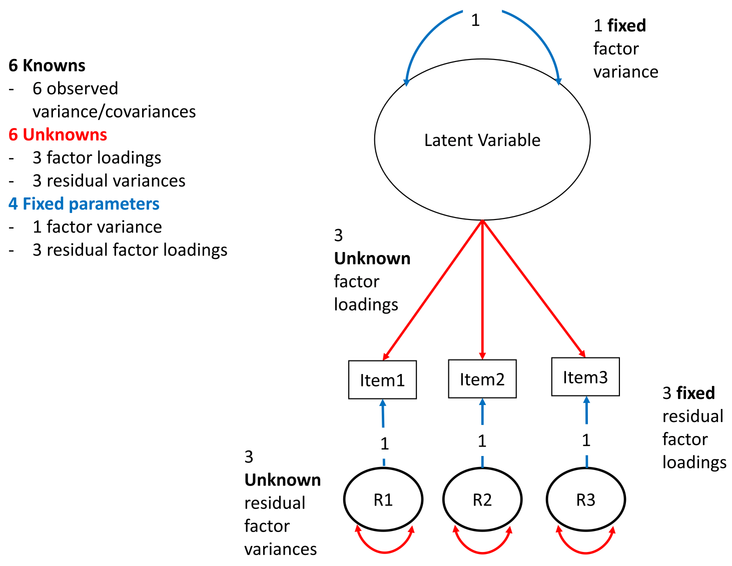
```
round(cov(Three.variables),2)
```

```
##      V1   V2   V3
## V1 1.06 0.36 0.45
## V2 0.36 1.04 0.62
## V3 0.45 0.62 0.99
```

- We have 6 unique elements (3 variances and 3 covariances)

Implications for CFA

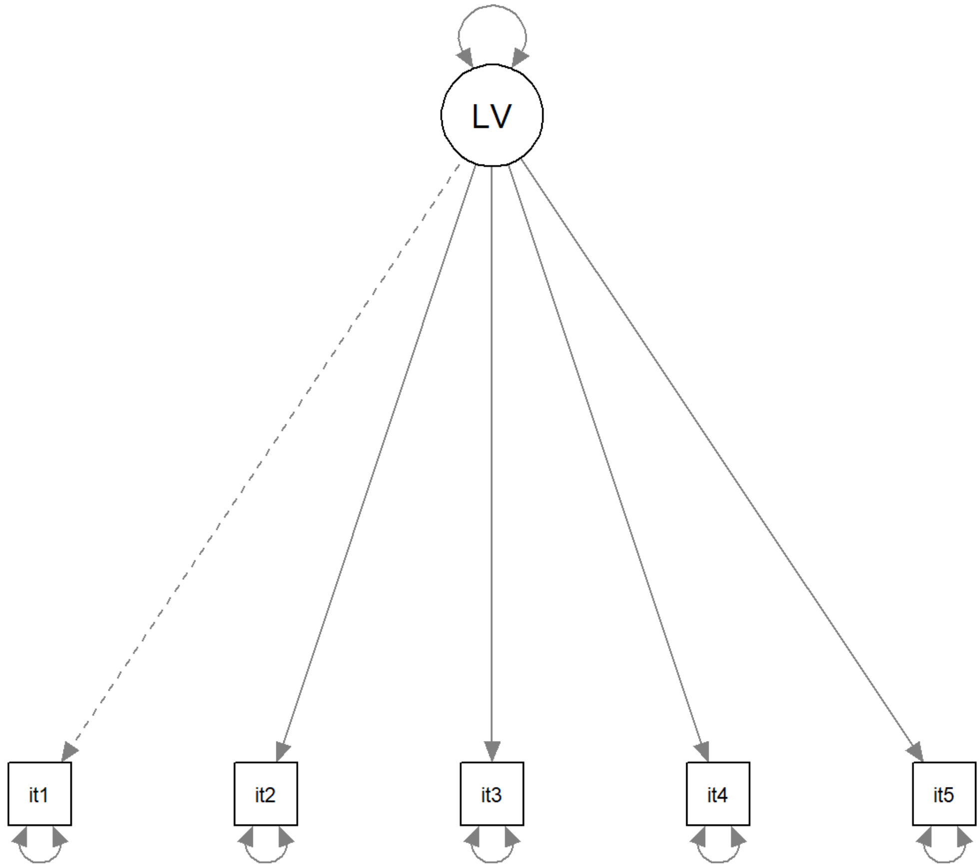
- We usually need a minimum of three observed variables for a just-identified model



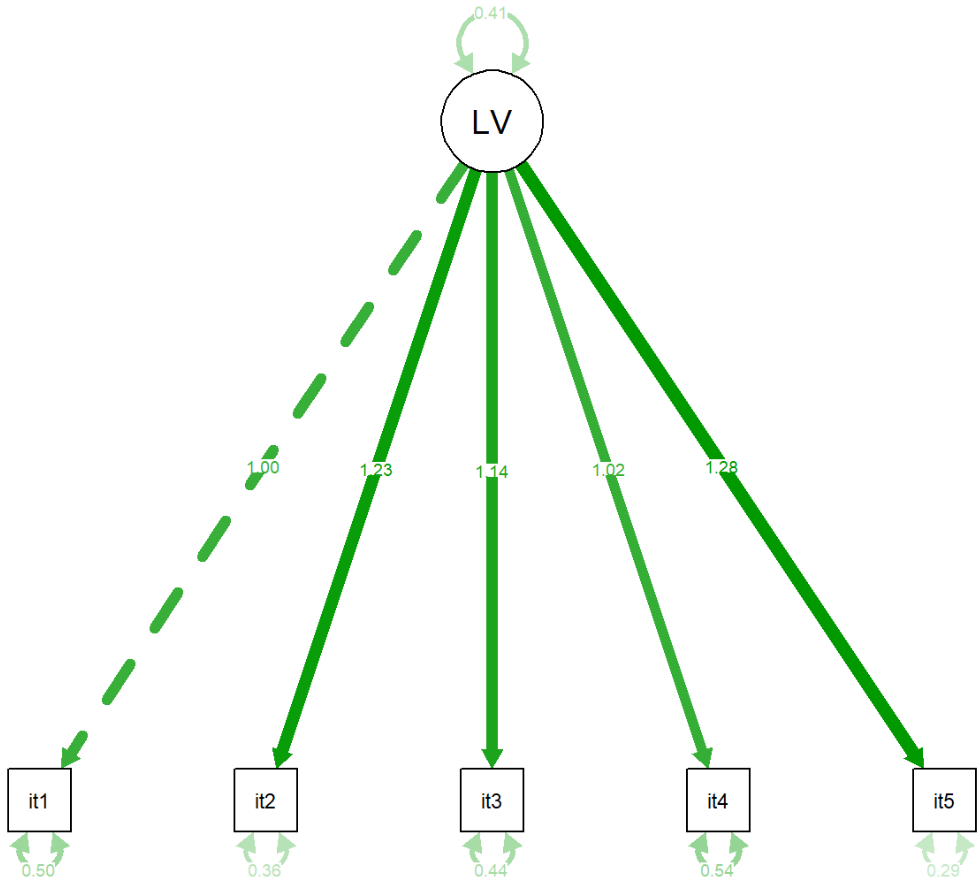
Model estimation

- After we have specified our model (& checked it is identified) we proceed to **estimation**
- Model estimation refers to finding the ‘best’ values for the unknown parameters

Specifying which parameters to estimate…



Finding the parameter values



Maximum likelihood estimation

- Maximum likelihood estimation is most commonly used
- Finds the parameters that maximise the likelihood of the data
- Begins with a set of starting values
- Iterative process of improving these values
  - *i.e. to minimise the difference between the sample covariance matrix and the covariance matrix implied by the parameter values*
- Terminates when the values are no longer substantially improved across iterations
  - *At this point* ***convergence*** *is said to have been reached*

No convergence?

- Sometimes estimation fails
- Common reasons are:
  - *The model is not identified*
  - *The model is very mis-specified*
  - *The model is very complex so more iterations are needed than the program default*

Maximum likelihood estimation assumptions

- Large sample size
- Multivariate normality
- Variables are on a continuous scale

Alternative estimators

- Robust maximum likelihood estimation
  - *For non-normal data*
- Weighted least squares, unweighted least squares or diagonally weighted least squares
  - *For ordinal data*

Model evaluation

- Once the model has been evaluated, we ask: *how good is the model?*
  - *Global fit*
  - *Local fit*

Global fit

- χ^2
  - *When we use maximum likelihood estimation we obtain a* χ^2 *value for the model*
  - *This can be compared to a* χ^2 *distribution with degrees of freedom equal to our model degrees of freedom*
  - *Statistically significant* χ^2 *suggests the CFA model does not do a good job of reproducing the observed variance-covariance matrix*
- However, χ^2 does not work well in practice
  - *Leads to the rejection of models that are only trivially mis-specified*

Alternatives to χ^2

- Absolute fit
  - *Standardised root mean square residual (**SRMR**)*
    - measures the discrepancy between the observed correlation matrix and model-implied correlation matrix
    - ranges from 0 to 1 with 0=perfect fit
    - values <.05 considered good
- Parsimony-corrected
  - *Corrects for the complexity of the model*
  - *Adds a penalty for having more degrees of freedom*
  - *Root mean square square error of approximation (**RMSEA**)*
    - 0=perfect fit
    - values <.05 considered good

Incremental fit indices

- Compares the model to a more restricted baseline model
  - *Usually an ‘independence’ model where all observed variable covariances fixed to 0*
- Comparative fit index (**CFI**)
  - *ranges between 0 and 1 with 1=perfect fit*
  - *values > .95 considered good*
- Tucker-Lewis index (**TLI**)
  - *includes a parsimony penalty*
  - *values >.95 considered good*

Local fit

- It is also possible to examine **local** areas of mis-fit
- **Modification indices** estimate the improvement in χ^2 that could be expected from including an additional parameter
  - *e.g., a cross-loading, residual covariance or latent variable covariance*
- **Expected parameter changes** estimate the value of the parameter were it to be included

Making model modifications

- Modification indices and expected parameter changes can be helpful for identifying how to improve the model
- However:
  - *Modifications should be made iteratively*
  - *Don't go overboard: may just be capitalising on chance*
  - *Make sure the modifications can be justified on substantive grounds*
  - *Be aware that this becomes an exploratory modeling practice*
  - *Ideally replicate the new model in an independent sample*

Other considerations in model evaluation

- Ideally:
  - *Factor loadings should be statistically significant*
  - *Standardised factor loadings should be >|.30|*
  - *Else some items/parameters could be trimmed from the model*
  - *(with the same caveats as on previous slide)*
- Check for **Heywood cases**
  - *Parameter estimates that are outside the permissable range*
  - *E.g., correlations >1, negative residual variances*
  - *May require modifications to the model to address*

Interpreting a CFA

- To aid interpretation we can request a fully **standardised solution**
- Converts loadings/covariances to a correlation metric
- Thereafter, the interpretation is similar to EFA:
  - *Loadings tell us strength of association between latent factor and items*
  - *Factor correlations tell us how strongly associated latent factors are*

Conducting a CFA model using lavaan

- Lavaan = **La**tent **Va**riable **An**alysis
- Used to specify and estimate latent variable models
- Three steps:

```
#step 1: specify the model

model<-'LV=~V1+V2+V3+V4'
  # we write the model using lavaan syntax enclosed in single quote marks

#step2: estimate the model

model.est<-cfa(model=model, data=df)
   # 'model= ' refers to a lavaan syntax object with the model specification
   # 'data= ' gives name of the dataframe in which to find the variables
#step3: inspect the results

summary(model.est)
    # the summary function shows us output from a fitted model
```

Model specification

- Specification uses lavaan syntax:

```
# simple regression model

Regression<-'DV~IV'

# multiple regression model

Multiple.regression<-'DV~IV1+IV2+IV3'

#covariance between two variables

Covariance<-'V1~~V2'

#latent factor specification

CFA<-'LV=~V1+V2+V3+V4'
```

Model specification for our aggression example

1. I hit someone
2. I kicked someone
3. I shoved someone
4. I battered someone
5. I physically hurt someone on purpose
6. I deliberately insulted someone
7. I swore at someone
8. I threatened to hurt someone
9. I called someone a nasty name to their face
10. I shouted mean things at someone

```
agg_m<-
  'Pagg=~item1+item2+item3+item4+item5

   Vagg=~item6+item7+item8+item9+item10

   Pagg~~Vagg'
```

Model estimation in lavaan

- To estimate the model, we then feed the object we just created into the cfa( ) function
- We also name the dataset containing the model
  - *Lavaan will compute the variance-covariance matrix internally*

```
agg_m.est<-cfa(agg_m, data=agg.items)
```

Scaling constraints

- By default, cfa( ) will scale the latent variables by fixing the first indicator for each latent factor to 1
- To override this and fix latent factor variances instead, we can write:

```
agg_m.est<-cfa(agg_m, data=agg.items, std.lv=T)
```

Model evaluation

- We can check the model fit using the summary( ) function:

```
summary(agg_m.est, fit.measures=T)
```

```
## lavaan 0.6-5 ended normally after 19 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         21
##
##   Number of observations                          1000
##
## Model Test User Model:
##
##   Test statistic                                53.494
##   Degrees of freedom                                34
##   P-value (Chi-square)                           0.018
##
## Model Test Baseline Model:
##
##   Test statistic                              4578.875
##   Degrees of freedom                                45
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.996
##   Tucker-Lewis Index (TLI)                       0.994
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -11707.817
##   Loglikelihood unrestricted model (H1)     -11681.070
##
##   Akaike (AIC)                               23457.633
##   Bayesian (BIC)                             23560.696
##   Sample-size adjusted Bayesian (BIC)        23493.999
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.024
##   90 Percent confidence interval - lower         0.010
##   90 Percent confidence interval - upper         0.036
##   P-value RMSEA <= 0.05                          1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.027
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   Pagg =~
##     item1             0.643    0.028   22.845    0.000
##     item2             0.790    0.028   28.635    0.000
##     item3             0.729    0.028   25.827    0.000
##     item4             0.656    0.029   22.561    0.000
```

```
##     item5           0.819    0.027   30.653    0.000
##   Vagg =~
##     item6           0.610    0.030   20.579    0.000
##     item7           0.849    0.025   33.903    0.000
##     item8           0.858    0.025   34.566    0.000
##     item9           0.647    0.029   22.392    0.000
##     item10          0.672    0.028   23.936    0.000
##
## Covariances:
##                  Estimate  Std.Err  z-value  P(>|z|)
##   Pagg ~~
##     Vagg            0.145    0.035    4.152    0.000
##
## Variances:
##                  Estimate  Std.Err  z-value  P(>|z|)
##    .item1           0.496    0.025   19.587    0.000
##    .item2           0.361    0.022   16.469    0.000
##    .item3           0.441    0.024   18.311    0.000
##    .item4           0.534    0.027   19.685    0.000
##    .item5           0.293    0.020   14.586    0.000
##    .item6           0.620    0.030   20.928    0.000
##    .item7           0.210    0.016   13.253    0.000
##    .item8           0.191    0.016   12.323    0.000
##    .item9           0.558    0.027   20.562    0.000
##    .item10          0.500    0.025   20.182    0.000
##     Pagg            1.000
##     Vagg            1.000
```

Model evaluation

- We can examine local mis-specifications using the modindices() function

```
modindices(agg_m.est, sort=T)
```

```
##       lhs op    rhs    mi    epc sepc.lv sepc.all sepc.nox
## 25   Pagg =~  item7 8.836  0.059   0.059    0.061    0.061
## 56  item3 ~~  item9 7.396  0.048   0.048    0.097    0.097
## 28   Pagg =~ item10 4.686 -0.055  -0.055   -0.057   -0.057
## 33   Vagg =~  item5 4.446 -0.047  -0.047   -0.048   -0.048
## 27   Pagg =~  item9 4.080  0.054   0.054    0.055    0.055
## 66  item5 ~~  item8 3.760 -0.022  -0.022   -0.092   -0.092
## 77  item8 ~~ item10 3.509  0.031   0.031    0.100    0.100
## 43  item2 ~~  item3 3.433  0.035   0.035    0.088    0.088
## 26   Pagg =~  item8 3.419 -0.036  -0.036   -0.037   -0.037
## 32   Vagg =~  item4 3.246  0.047   0.047    0.048    0.048
## 51  item3 ~~  item4 3.182 -0.034  -0.034   -0.070   -0.070
## 24   Pagg =~  item6 3.106 -0.049  -0.049   -0.049   -0.049
## 29   Vagg =~  item1 2.680  0.041   0.041    0.043    0.043
## 46  item2 ~~  item6 2.427 -0.027  -0.027   -0.057   -0.057
## 67  item5 ~~  item9 2.137  0.023   0.023    0.057    0.057
## 62  item4 ~~  item9 2.034 -0.027  -0.027   -0.049   -0.049
## 75  item7 ~~ item10 1.970 -0.023  -0.023   -0.071   -0.071
## 69  item6 ~~  item7 1.726  0.022   0.022    0.060    0.060
## 61  item4 ~~  item8 1.657  0.017   0.017    0.054    0.054
## 42  item1 ~~ item10 1.592  0.022   0.022    0.044    0.044
## 68  item5 ~~ item10 1.506 -0.018  -0.018   -0.048   -0.048
## 45  item2 ~~  item5 1.427 -0.024  -0.024   -0.074   -0.074
## 65  item5 ~~  item7 1.191  0.012   0.012    0.050    0.050
## 49  item2 ~~  item9 1.182 -0.018  -0.018   -0.040   -0.040
## 47  item2 ~~  item7 1.134  0.013   0.013    0.046    0.046
## 58  item4 ~~  item5 0.970  0.018   0.018    0.047    0.047
## 70  item6 ~~  item8 0.919 -0.016  -0.016   -0.046   -0.046
## 54  item3 ~~  item7 0.908 -0.012  -0.012   -0.040   -0.040
## 57  item3 ~~ item10 0.854 -0.016  -0.016   -0.033   -0.033
## 64  item5 ~~  item6 0.847 -0.015  -0.015   -0.036   -0.036
## 60  item4 ~~  item7 0.526  0.010   0.010    0.029    0.029
## 34  item1 ~~  item2 0.503 -0.013  -0.013   -0.031   -0.031
## 37  item1 ~~  item5 0.497  0.013   0.013    0.034    0.034
## 53  item3 ~~  item6 0.469  0.013   0.013    0.024    0.024
## 38  item1 ~~  item6 0.368  0.012   0.012    0.021    0.021
## 39  item1 ~~  item7 0.260  0.007   0.007    0.021    0.021
## 35  item1 ~~  item3 0.259 -0.009  -0.009   -0.020   -0.020
## 72  item6 ~~ item10 0.183 -0.008  -0.008   -0.015   -0.015
## 30   Vagg =~  item2 0.159 -0.009  -0.009   -0.009   -0.009
## 40  item1 ~~  item8 0.143 -0.005  -0.005   -0.016   -0.016
## 76  item8 ~~  item9 0.126 -0.006  -0.006   -0.018   -0.018
## 63  item4 ~~ item10 0.118 -0.006  -0.006   -0.012   -0.012
## 73  item7 ~~  item8 0.113 -0.007  -0.007   -0.037   -0.037
## 50  item2 ~~ item10 0.080 -0.004  -0.004   -0.011   -0.011
## 41  item1 ~~  item9 0.078 -0.005  -0.005   -0.010   -0.010
## 36  item1 ~~  item4 0.073  0.005   0.005    0.010    0.010
## 55  item3 ~~  item8 0.056 -0.003  -0.003   -0.010   -0.010
## 48  item2 ~~  item8 0.049  0.003   0.003    0.010    0.010
## 71  item6 ~~  item9 0.043  0.004   0.004    0.007    0.007
## 31   Vagg =~  item3 0.018  0.003   0.003    0.003    0.003
## 44  item2 ~~  item4 0.018  0.003   0.003    0.006    0.006
## 59  item4 ~~  item6 0.018  0.003   0.003    0.005    0.005
## 78  item9 ~~ item10 0.001  0.001   0.001    0.001    0.001
## 52  item3 ~~  item5 0.001  0.000   0.000    0.001    0.001
## 74  item7 ~~  item9 0.000  0.000   0.000    0.000    0.000
```

Standardised parameter estimates

- We can also inspect the standardised parameter estimates

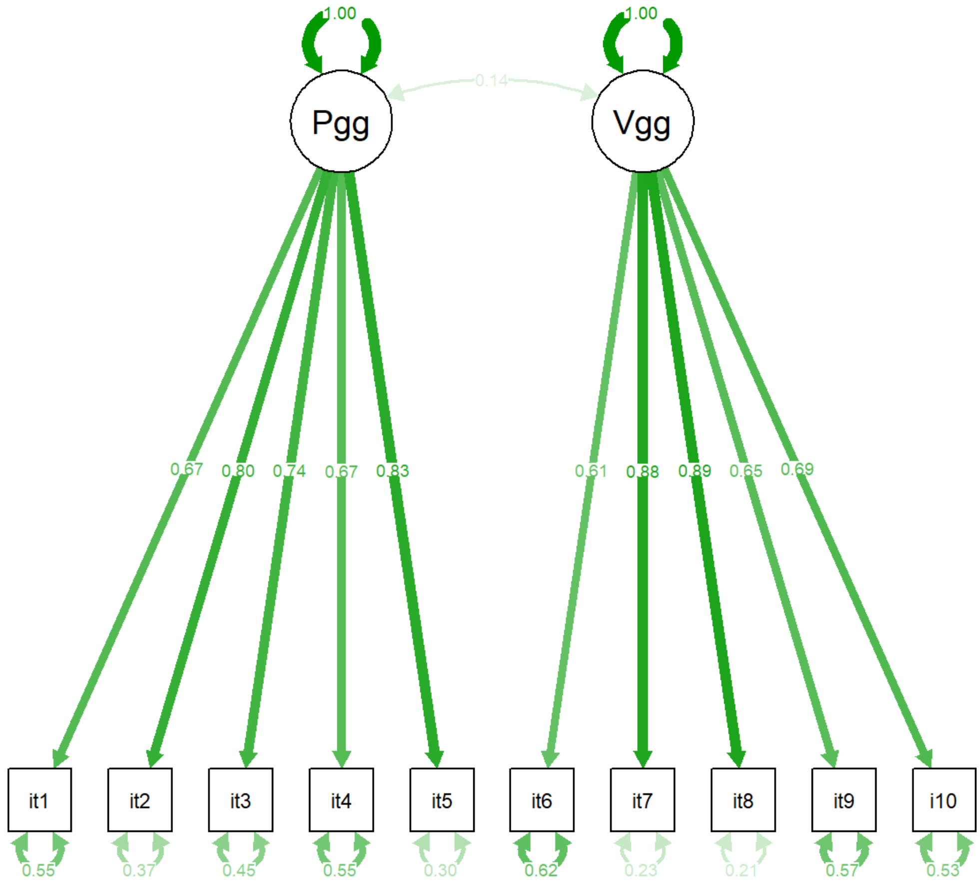
```r
summary(agg_m.est, standardized=T)
```

```
## lavaan 0.6-5 ended normally after 19 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         21
##
##   Number of observations                          1000
##
## Model Test User Model:
##
##   Test statistic                                53.494
##   Degrees of freedom                                34
##   P-value (Chi-square)                           0.018
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Pagg =~
##     item1             0.643    0.028   22.845    0.000    0.643    0.674
##     item2             0.790    0.028   28.635    0.000    0.790    0.796
##     item3             0.729    0.028   25.827    0.000    0.729    0.739
##     item4             0.656    0.029   22.561    0.000    0.656    0.668
##     item5             0.819    0.027   30.653    0.000    0.819    0.834
##   Vagg =~
##     item6             0.610    0.030   20.579    0.000    0.610    0.613
##     item7             0.849    0.025   33.903    0.000    0.849    0.880
##     item8             0.858    0.025   34.566    0.000    0.858    0.891
##     item9             0.647    0.029   22.392    0.000    0.647    0.655
##     item10            0.672    0.028   23.936    0.000    0.672    0.689
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Pagg ~~
##     Vagg              0.145    0.035    4.152    0.000    0.145    0.145
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .item1             0.496    0.025   19.587    0.000    0.496    0.545
##    .item2             0.361    0.022   16.469    0.000    0.361    0.366
##    .item3             0.441    0.024   18.311    0.000    0.441    0.454
##    .item4             0.534    0.027   19.685    0.000    0.534    0.554
##    .item5             0.293    0.020   14.586    0.000    0.293    0.304
##    .item6             0.620    0.030   20.928    0.000    0.620    0.625
##    .item7             0.210    0.016   13.253    0.000    0.210    0.226
##    .item8             0.191    0.016   12.323    0.000    0.191    0.206
##    .item9             0.558    0.027   20.562    0.000    0.558    0.571
##    .item10            0.500    0.025   20.182    0.000    0.500    0.525
##     Pagg              1.000                               1.000    1.000
##     Vagg              1.000                               1.000    1.000
```

Visualising the model

- Sempaths() from the semPlot package can be used to visual a model as a SEM diagram

```
semPaths(agg_m.est, what='stand')
```



Writing up a CFA model

- Methods
  - *Model(s) being tested*
  - *Scaling /identifcation method*
  - *Estimation method*
  - *Criteria that used to judge fit*
- Results
  - *Model fit (χ^2 test, CFI, TLI, RMSEA, SRMR)*
  - *Any modifications made and why*
  - *Model parameters (in a SEM diagram or table)*

Cautions regarding CFA

- Good fit doesn’t guarantee that the model is ‘correct’
- Be careful about ‘reifying’ latent variables
- Even when there are no common factors, CFA models can fit well

Summary

CFA involves testing a hypothesised factor structure

- Specifying a model
  - *Identification and scaling*
- Estimating that model
  - *e.g., maximum likelihood estimation*
- Seeing how well that model fits the data
  - *Global and local fit*
- Interpreting the model